**Chapter 3**: **Proposed Methodology**

**Content:**

3.1. System Design

3.1.1. Introduction

3.1.2. Block Diagram

3.1.3. System architecture diagram

3.1.4. Data Flow Diagram

3.1.5. Software Design Approach

3.2. Time Line Chart

3.3. Gantt Chart

3.4. Conclusion

**3.1. System Design**

**3.1.1. Introduction**

Systems design is a methodical approach to crafting a system. This can be carried out in a top-down or bottom-up manner, but in both cases, the process is systematic and considers all the interconnected aspects of the system being created. This includes everything from the system's structure, the necessary hardware and software, right down to the management and transformation of data as it traverses the system.

Systems design is closely intertwined with systems analysis, systems engineering, and systems architecture. The origins of the systems design approach can be traced back to the period just before World War II when engineers were grappling with intricate problems related to control and communication. They recognized the need to formalize their work into a structured discipline with established methods, especially given the emergence of new fields like information theory, operations research, and computer science.

System design is the process of defining the various elements of a system, encompassing its architecture, modules, components, the interfaces between these components, and the flow of data within the system. Its primary aim is to meet the specific needs and requirements of a business or organization by engineering a coherent and efficiently functioning system.

**3.1.2. Block Diagram**

A block diagram is a visual tool that simplifies complex systems or concepts by using labeled blocks to represent individual or multiple elements, entities, or ideas. These blocks are connected by lines to illustrate connections or associations among them. An entity relationship diagram (ERD), which is a specific type of block diagram, depicts an information system by showcasing the relationships between people, objects, places, concepts, or events within that system.

Block diagrams find extensive application in engineering, where they are essential for creating diagrams in fields such as electronics, hardware, software, and various processes. Typically, they offer a high-level, less detailed perspective on concepts and systems. These diagrams are particularly valuable for diagnosing and resolving technical problems.

**3.1.3. System architecture diagram**

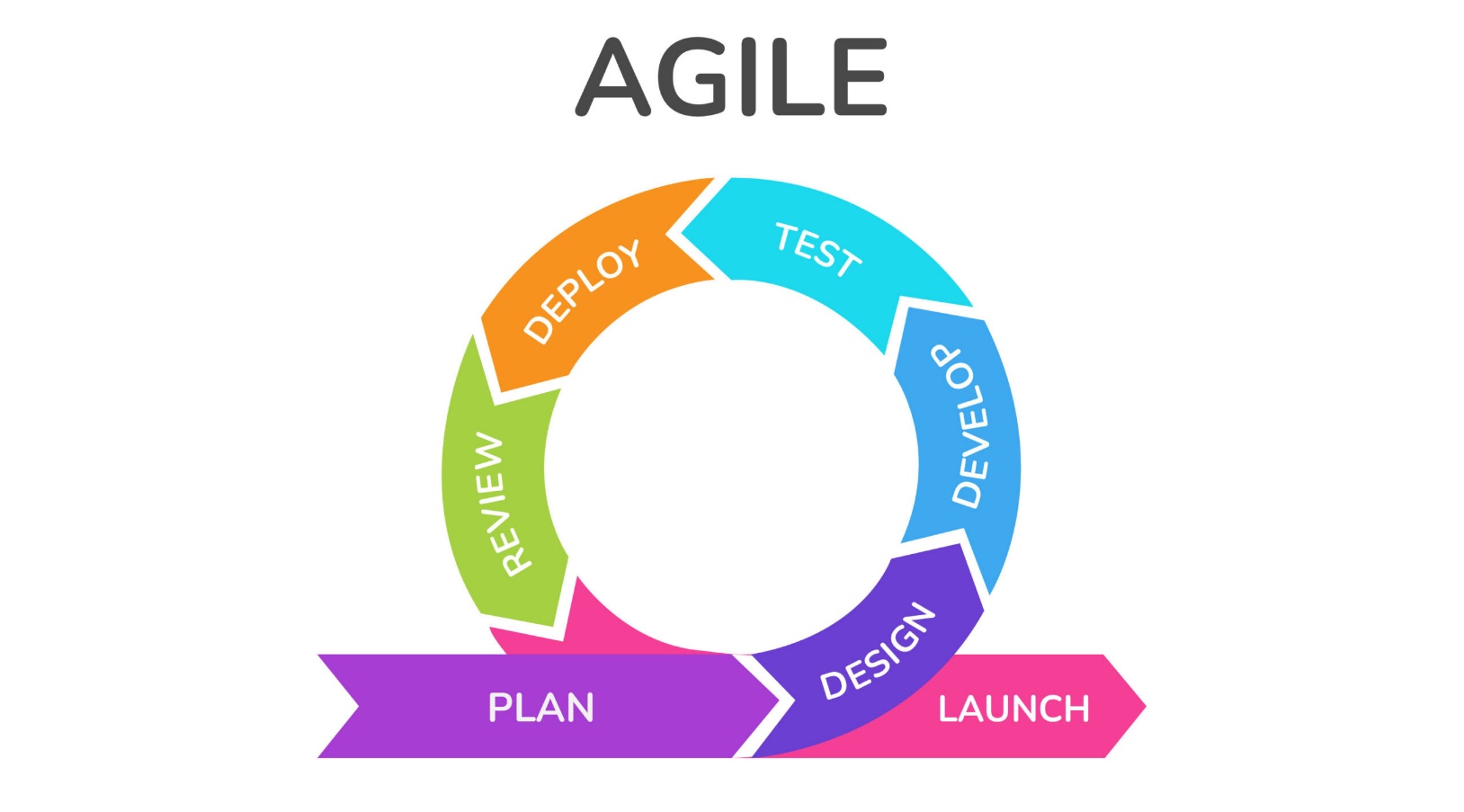
A system architecture diagram serves as a visual representation of how different components interact within a system. Typically, these diagrams are designed for systems that encompass both hardware and software elements, depicting their relationships. However, they can also be tailored for web applications. In the context of a web application, the system architecture design would encompass components like the database, application server, web server, internet, browser, and potentially other relevant elements. The level of detail in these diagrams can vary, offering a higher-level, less intricate overview. These visual representations play a crucial role in troubleshooting technical issues.

**3.1.4. Data Flow Diagram**

Data Flow Diagrams, commonly referred to as DFDs, serve as visual representations of data flow within a business information system. They depict the processes responsible for transferring data from input sources to file storage and report generation. DFDs can be categorized into two types: logical and physical. The logical data flow diagram illustrates the path of data within a system to execute specific business functions, while the physical data flow diagram outlines how the logical data flow is put into practice.

**3.1.5. Software Design Approach**

An Agile process model is a software development methodology that emphasizes flexibility, collaboration, and iterative development. It is characterized by breaking the project into small, manageable increments, allowing for regular reassessment and adaptation based on feedback. Agile methodologies prioritize customer satisfaction, rapid delivery, and the ability to respond to changing requirements.



**The Agile Software Development process typically consists of the following steps:**

1. **Requirements Gathering:** The customer’s requirements for the software are gathered and prioritized.
2. **Planning:** The development team creates a plan for delivering the software, including the features that will be delivered in each iteration.
3. **Development:** The development team works to build the software, using frequent and rapid iterations.
4. **Testing:** The software is thoroughly tested to ensure that it meets the customer’s requirements and is of high quality.
5. **Deployment:** The software is deployed and put into use.
6. **Maintenance:** The software is maintained to ensure that it continues to meet the customer’s needs and expectations.

**Principles of Agile Software Development Process are:**

1. Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. It welcomes changing requirements, even late in development.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shortest timescale.
4. Build projects around motivated individuals. Give them the environment and the support they need, and trust them to get the job done.
5. Working software is the primary measure of progress.
6. Simplicity the art of maximizing the amount of work not done is essential.
7. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

**Development in Agile:** Let’s see a brief overview of how development occurs in Agile philosophy.

* In Agile development, Design and Implementation are considered to be the central activities in the software process.
* Design and Implementation phase also incorporates other activities such as requirements elicitation and testing into it.
* In an agile approach, iteration occurs across activities. Therefore, the requirements and the design are developed together, rather than separately.
* The allocation of requirements and the design planning and development as executed in a series of increments. In contrast with the conventional model, where requirements gathering needs to be completed in order to proceed to the design and development phase, it gives Agile development an extra level of flexibility.
* An agile process focuses more on code development rather than documentation.

**3.2. Time Line Chart**



**3.3. Gantt Chart**



**3.4. Conclusion**

In conclusion, Agile methodologies are the perfect fit for the "3D Outfit Customizer" project in the fashion industry. Let's break down why Agile is the way to go: First off, Agile's customer-centric approach is spot-on for a project catering to the diverse and ever-evolving tastes of fashion enthusiasts. By involving users in the development process, the "3D Outfit Customizer" can be tailored to meet individual needs and unique styles. The fashion industry's fast-paced nature requires adaptability, and Agile delivers. Its flexibility enables the project team to swiftly respond to shifting trends and requirements, ensuring the platform's continued relevance and competitiveness. Agile's iterative development approach is another big win. It means we can get valuable features into users' hands earlier, allowing them to start creating and customizing outfits sooner, enhancing the overall experience. Open and transparent communication is at the core of Agile. This is vital for understanding and adapting to the ever-changing world of fashion, ensuring the project stays in sync with customer needs. Last but not least, Agile's emphasis on quality assurance is a big deal. It ensures the final product meets high standards, which is a critical factor for success in the fashion industry.

So, in a nutshell, Agile's customer-centric, adaptable, and collaborative approach, coupled with its focus on risk management and quality assurance, makes it the top choice for the "3D Outfit Customizer" project. Going Agile will help us tackle the unique challenges of the fashion industry and deliver a more personalized, competitive, and high-quality solution to our users.